REMARKS

Applicants respectfully request that the above-identified application be reexamined.

The April 11, 2007, Office Action ("Office Action") rejected all of the claims in this application. More specifically, Claim 7 was rejected under 35 U.S.C. § 112 as being based on insufficient antecedent basis for a limitation in this claim. The language of Claim 7 has been amended to provide sufficient antecedent basis for the limitation. Thus, Claim 7 will not be further discussed. Claims 6, 15-17, and 21 were rejected under 35 U.S.C. § 102(b) as being unpatentable over U.S. Patent No. 5,220,675 ("Padawer et al."). Claims 11-14 were rejected under 35 U.S.C. § 102(b) as being unpatentable over U.S. Patent No. 5,625,783 ("Ezekiel et al."). Claims 1-5, 7-8, 18-20, and 22-24 were rejected 35 U.S.C. § 103(a) as being unpatentable over Padawer et al. in view of Ezekiel et al. Claims 9 and 10 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Padawer et al. and Ezekiel et al. in view of U.S. Patent No. 6,357,038 ("Scouten").

While applicants respectfully disagree with the rejections of the claims, in order to advance the prosecution of this application, various clarifying amendments have been made to the language of the claims. Applicants respectfully submit that all the claims in this application are clearly allowable in view of the cited and applied references.

Prior to discussing in detail why applicants believe that all of the claims in this application are allowable, a brief description of the disclosed subject matter and brief descriptions of the teachings of the cited and applied references are provided. The following discussions of the disclosed subject matter and the cited and applied references are not provided to define the scope or interpretation of any of the claims of this application. Instead, these discussions are provided to help the United States Patent and Trademark Office better appreciate important claim distinctions discussed thereafter.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

Disclosed Subject Matter

Many software applications ("applications") provide user interfaces (UIs) for users to use to invoke the functions of the software applications. Often, the functions are represented in the UI as selectable items, including text and image, or both. The selectable items are often organized into a hierarchy of lists and sublists. Each selectable item has an "insert location" in a list or sublist. An "insert location" is the preferred location of the selectable item in the list or sublist. Some applications allow selectable items to be added to the applications while the applications are running. Lists and sublists of selectable items are added to an application's UI to provide access to added functions. It is also possible to remove selectable items and/or lists and sublists of selectable items in order to block access to related functions.

Additional functions are usually contained in extension files ("extensions"). Because additional function selectable items each have an insert location in a list or sublist, when multiple extensions are added to a host application overlaps and conflicts can occur between the insert locations of the selectable items associated with the extensions. The disclosed improved programming interface allows multiple extensions to be used to modify the host application's UI without causing overlaps and conflicts between the selectable items.

In the improved programming interface, functions are represented by command items. A host application passes the available insert locations to extensions that seek to modify the UI. Each extension returns a set of command items that represent the functions that the extension requests added to the host's UI at one or more of the available insert locations. The host loads the extensions and recursively modifies the UI as requested by each extension in accordance with the available insert locations and the extensions load order relative to the other extensions. Load order may be arbitrary or predefined.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

The host application constrains the extensions by permitting new commands to be inserted in the host application's UI only at certain insert locations. An insert location represents an actual location in the host application's UI where the inserted command's extension UI will be displayed. The actual location of an insert location in a host application UI is dynamically defined by the host application and may be changed and even be eliminated from one version of the host application UI to another. Because the host application dynamically defines the insert locations, no user input is required to insert menus and menu items for additional functions.

U.S. Patent No. 5,220,675 (Padawer et al.)

Padawer et al. describes enabling a user to customize a menu while a computer program is executing by manually creating a menu item and associating the menu item with an external computer program. Padawer et al. is manual and does not enable menu items for individual functions to be included in a menu.

U.S. Patent No. 5,625,783 (Ezekiel et al.)

Ezekiel et al. describes providing different sets of menus in one single computer program depending on the software components of the one single computer program that are active. In response to actions taken by users or to changes in the operating mode of the computer program, new menu items from external software components may be inserted into the existing menus to provide additional functions.

U.S. Patent No. 6,357,038 (Scouten)

Scouten describes "macros." A macro is a computer program expressed as a script of a series of actions. A simple example of a macro is a script of a series of keystrokes that invoke actions. The actions, e.g., keystrokes, are recorded by an application in a macro. The macro can then be read later by the application to replay the series of keystrokes and invoke the actions.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

Scouten focuses on cross platform and cross operating system macros that are embedded in macro files that include a plurality of versions of executable configuration instructions, one version for each platform and/or operating system. A macro file has an identifier that associates the macro file with the application that was used to produce the macro file and is also used to invoke the application to replay the actions in the macro. Scouten does not enable lists and sublists of selectable items, e.g., menus, submenus, and menu items, that can be inserted into the application's UI.

Rejection of Claims 6, 15-17 and 21 Under 35 U.S.C. § 102(b) Based on Padawer et al.

Remarks accompanying the rejection of Claim 6 in the Office Action read as follows:

> Padawer et al. disclose a method for the method comprising:
>
> **loading at least one extension** (i.e. ... item 1218 **updates the CMI array with values returned from Add_Atom** ... col 7 lines 10-15 and Fig. 12);
>
> uniquely identifying for the at least one extension an available insert location where the extension may request to insert commands into a host UI (i.e. ... whether the CMI array location or slot is available ... col 6 lines 61-65 and Fig. 12); and
>
> obtaining from the at least one extension at least one uniquely identified command to insert at the available insert location; and integrating the command in the available insert location in accordance with the extension's load order. (i.e. ... see flow diagram for Add_Atom routine ... col 7 lines 19-29 and Fig. 13). (Emphasis added.)

Applicants respectfully disagree that updating the CMI array with values returned from Add_Atom, as recited in Padawer et al., is the same as "loading at least one extension," as recited in Claim 6. As currently amended, Claim 6 reads:

> 6. A method for modifying a host user interface (UI), the method comprising:
>
> **loading at least one extension, an extension being an application capable of extending the functionality of the host**;
>
> uniquely identifying for the at least one extension an available insert location where the extension may request to insert commands into a host UI;

obtaining from the at least one extension at least one uniquely identified command to insert at the available insert location; and

integrating the command in the available insert location in accordance with the extension's load order. (Emphasis added.)

An extension, as more clearly recited in amended Claim 6, is an application capable of extending the functionality of the host. In contrast, Padawer et al. relies on atoms. Atoms, as disclosed in lines 14-16 of Col. 5 of Padawer et al., are "handles to an atom table. This atom table is used to store and retrieve desired strings." A string is generally defined as a data structure composed of a sequence of characters usually representing human-readable text. Strings and tables of strings are not extensions. They cannot extend the functionality of a host. Hence, Padawer et al. does not disclose, teach, or remotely suggest the subject matter of Claim 6. As a result, applicants respectfully submit that Claim 6 is allowable.

Remarks accompanying the rejection of Claim 15 in the Office Action read as follows:

Padawer et al. disclose a system to safely modify a host user interface (UI) with extensions, the system comprising:

a memory in which to store a command item representing a UI for a command (i.e. . . . array 1002 is used to store a an array of command menu items . . . col 5 lines 6-9 and Fig. 10A);

and an insert location available to integrate the command into a host UI (i.e. . . . Determine whether a CMI array location or slot is available . . . col 6 lines 61-65 and Fig. 12);

a processing unit operable to return the command item in response to request for commands to integrate into the host UI at the insert location (i.e. . . . a flow diagram to "Add" utility 1128 which is invoked by the routine 1110 to create a new entry in the CMI array 1002 . . . col 6 lines 59-61 and Fig. 12); and

a display device to display the command integrated into the host UI at the insert location in accordance with command item UI (i.e. . . . displaying the user customizable interface . . col 2 lines 37-39 and Fig. 1-9). (Emphasis added.)

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

Applicants respectfully submit that an array of command menu items as recited in Padawer et al. is not the same as a memory in which to store a command item representing a UI for a command from an extension. As amended, Claim 15 reads as follows:

> 15. A system to safely modify a host user interface (UI) with extensions, the system comprising:
>
> **a memory component for storing a command item associated with a command from an extension, an extension being an application capable of extending the functionality of the host UI,** and an insert location available to integrate the command into a host UI;
>
> a processing unit operable to return the command item in response to a request for commands to integrate into the host UI at the insert location; and
>
> a display device to display the command from the extension integrated into the host UI at the insert location in accordance with command item UI. (Emphasis added.)

Lines 6-9 in Col. 5 of Padawer et al. read as follows:

> Specifically, array 1002 is used to store an array of command menu items CMIs wherein each CMI array entry corresponds to a particular menu item.

The command items of Padawer et al. are not associated with commands from extensions as recited in Claim 15. Further, as explained above in the section regarding Claim 6, Padawer et al. does not disclose using extensions. Because the command items of Padawer et al. are not associated with commands from extensions and because Padawer et al. does not disclose using extensions, Padawer et al. does not disclose a memory component for storing command items associated with commands from extensions. Thus, Padawer et al. does not disclose, teach, or remotely suggest the subject matter of Claim 15. As a result, applicants respectfully submit that independent Claim 15 is allowable. Applicants further submit that because dependent Claims 16 and 17 depend from independent Claim 15, Claims 16 and 17 are allowable for at least the reasons that independent Claim 15 is allowable.

Remarks accompanying the rejection of Claim 21 read as follows:

> Padawer et al. discloses components for safely modifying a host user interface with an extension user interface, the medium comprising:
>
> a user interface (UI resource having a command item data structure in which to store a command UI, and an insert location data structure to store an available insert location in a host UI (i.e. . . array 1002 is used to store a (sic) an array of command menu items . . . col 5 lines 6-9 and Fig. 10A;
>
> and a host interface to expose the available insert locations to extensions that have commands to insert into the host UI (i.e. . . Determine whether a CMI array location or slot is available . . . col 6 lines 61-65 and Fig. 12):
>
> **and to receive a count of command items and command items representing the extension's commands (i.e. . . . determines the number of menu items . . . col 6 35-37 and Fig. 11);**
>
> an extension interface to receive the available insert locations from the host and to provide the count of command items and command items representing the extension's command (i.e. . . . Determine whether a CMI array location or slot is available . . . col 6 lines 61-65 and Fig. 12); and
>
> a host process to integrate the extension's command into the host interface in accordance with the UI resource (i.e. . . . a flow diagram to "Add" utility 1128 which is invoked by the routine 1110 to create a new entry in the CMI array 1002 . . . col 6 lines 59-61 and Fig. 12). (Emphasis added.)

Applicants respectfully submit that determining a number of menu items as recited in Padawer et al. is not the same as receiving a count of command items and command items representing an extension's commands as recited in Claim 21. Claim 21 reads as follows:

> 21. A computer-accessible medium having components for safely modifying a host user interface with an extension user interface, the medium comprising:
>
> a user interface (UI) resource having a command item data structure in which to store a command UI, and an insert location data structure to store an available insert location in a host UI; and
>
> a host interface to expose the available insert locations to extensions that have commands to insert into the host UI, and **to receive a count of command items and command items representing the extension's commands, an extension being an application capable of extending the functionality of the host UI**;
>
> an extension interface to receive the available insert locations from the host and to provide the count of command items and command items representing the extension's commands; and

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

> a host process to integrate the extension's command into the host interface in accordance with the UI resource. (Emphasis added.)

Lines 35-37 in Col. 6 of Padawer et al. read as follows:

> Once invoked, item 1102 determines the number of menu items in the CMI array and saves the determined number as the variable "count".

The menu items in the CMI array of Padawer et al. do not represent extension commands as recited in Claim 21. The number of menu items in a CMI array is not the same as a count of command items as recited in Claim 21. Further, as explained above in the section regarding Claim 6, Padawer et al. does not disclose using extensions. Because menu items in the CMI array of Padawer et al. do not represent extension commands and because Padawer et al. discloses using extensions, Padawer et al. does not receive a count of command items and command items representing an extension's commands. Thus, Padawer et al. does not disclose, teach, or remotely suggest the subject matter of Claim 21. As a result, applicants respectfully submit that independent Claim 21 is allowable.

Rejection of Claims 11-14 under 35 U.S.C. § 102(b) Based on Ezekiel et al.

Remarks accompanying the rejection of Claim 11 in the Office Action read as follows:

> Ezekiel et al. discloses a method of communicating between a host and an extension, the method comprising:
>
> an extension storing a UI for a command in a command item in preparation for communicating with a host (i.e. . . . each entry of command table can contain, for example, name of the command, . . . and a unique identifier . . . col 6 lines 34-43);
>
> a host issuing a call to the extension to return a number of command items to be inserted into an insert location (i.e. see the process of Shell-Host- which does similar steps in creating and display the new menu items form the registered command items. Col 9 lines 1-13 and Fig. 6;
>
> the extension returning the number of command items to be inserted into the insert location (i.e. . . . see package registration process . . . col 8 lines 12-28 and Fig. 4;
>
> the host issuing a call to an extension to return the command items to be inserted into the insert location (i.e. see the process of Shell- which does

> similar steps in creating and display the new menu items form the registered command items. Col 9 lines 1-13 and Fig. 6);
>
> the extension returning the command items to be inserted into the insert location (i.e. . . . see package registration process . . . col 8 lines 12-28 and Fig. 4); and
>
> the host integrating the command for each of the returned command items into a host UI in accordance with each command item's stored UI. (i.e. see the process of Shell-Host- which does similar steps in creating and display the new menu items form the registered command items. Col 9 lines 1-13 and Fig. 6).

Applicants respectfully disagree that the package registration process recited in lines 12-28 of Col. 8 in Ezekiel et al. and illustrated in Figure 4 of Ezekiel et al. is the same as an extension returning a number of command items to be inserted into an insert location as recited in Claim 11. It is clear in Figure 4 that the package registration process of Ezekiel et al. involves looping over each package, packet and template. However, in Claim 11, an extension only needs to return the number of command items to be inserted at a location. This is clearly an advantage because the host has the option of accepting or rejecting the command items depending on factors like the insert spaces available without looping through every package, packet and template. Hence, Ezekiel et al. does not disclose, teach, or remotely suggest the subject matter of Claim 11. As a result, applicants respectfully submit that independent Claim 11 is allowable. Applicants further submit that because dependent Claims 12-14 depend from independent Claim 11, Claims 12-14 are allowable for at least the reasons that independent Claim 11 is allowable.

Rejection of Claims 1-5, 7-8, 18-20, and 22-24 Under 35 U.S.C. § 103(a) Based on Padawer et al. in View of Ezekiel et al.

The remarks in the Office Action regarding the rejection of Claim 1 state that Padawer et al. does not disclose a method for uniquely identifying command items as recited in Claim 1. Applicants respectfully agree. The remarks in the Office Action go on to state that Ezekiel et al.,

in Col. 6, lines 33-43, discloses a method for uniquely identifying command items and that, therefore, it would have been obvious to one having ordinary skill in the art at the time of this invention to add a unique identifier for CMI array entries. Applicants respectfully disagree. As explained above in the section regarding the rejection of Claim 6, updating a CMI array with values returned from Add_Atom as recited in Padawer et al. is not the same as loading at least one extension as recited in Claim 6. Hence, adding unique identifiers for CMI array entries is not the same as providing unique identifiers for extensions. Thus, it would not have been obvious to one having ordinary skill in the art at the time of this invention to add a unique identifier for extensions. As a result, applicants respectfully submit that independent Claim 1 is allowable. Applicants further respectfully submit that dependent Claims 2-5 that depend directly or indirectly from independent Claim 1 are allowable for at least the reasons why Claim 1 is allowable.

The remarks in the Office Action regarding the rejection of Claim 7, which depends from Claim 1, state that Padawer et al. and Ezekiel et al. disclose a method for modifying a host user interface as in Claim 1. Regarding the additional language of Claim 7, the Office Action states that Ezekiel et al., in Col. 6, lines 33-43, discloses a method for uniquely identifying command items and that it would have been obvious to one having ordinary skill in the art at the time of this invention to add a unique identifier for CMI array entries. Applicants respectfully disagree. More importantly, Ezekiel et al. does not make up for the differences of Padawer et al. discussed above. Thus, even if Ezekiel et al. was combinable with Padawer et al., which applicants deny, the resulting combination would not meet the recitations of Claims 1 and 7 when combined. As a result, applicants respectfully submit that Claim 7 is allowable. Applicants further respectfully submit that dependent Claim 8, which that depends directly Claim 7, is allowable for at least the reasons why Claim 7 is allowable.

The remarks in the Office Action regarding the rejection of Claim 18, which depends from Claim 17, state that Padawer et al. does not disclose storing a universally unique identifier (UUID) with the insert location and command item as recited in Claim 18. Applicants respectfully agree. The Office Action remarks go on to state that Ezekiel et al., in Col. 6, lines 33-43, discloses a system that stores a UUID with the insert and command item and that it would have been obvious to one having ordinary skill in the art at the time of this invention to store a unique identifier for each command and insert locations. Applicants respectfully disagree. The cited lines of Ezekiel et al. do not disclose, teach or remotely suggest **universal** unique identifiers. Rather, Ezekiel et al. discloses identifiers that are locally, i.e., within each command table, unique. Hence, it would not have been obvious to one having ordinary skill in the art at the time of this invention to store a universal unique identifier for each command and insert location. Further, even if Ezekiel et al. did disclose what the remarks allege Ezekiel et al. discloses, which applicants deny, and even if it were obvious to combine Ezekiel et al. with Padawer et al., which applicants also deny, since Ezekiel et al. does not make up for the deficiencies of Claim 15, the independent claim from which Claim 18 depends, the resulting combination would still not meet the recitations of Claim 18 in combination with the claims from which Claim 18 depends. As a result, applicants respectfully submit that dependent Claim 18 is allowable. Applicants further respectfully submit that amended dependent Claims 19 and 20, which, as amended, now depend directly from Claim 18, are allowable for at least the reasons why Claim 18 is allowable.

The remarks in the Office Action regarding the rejections of Claim 22 and 23, which depend from Claim 21, are based on reasons similar to the reasons for the rejection of Claim 18. Hence, applicants further respectfully submit that Claims 22 and 23 are allowable for at least the reasons that Claim 21 and Claim 18 are allowable.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS[PLLC]
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

The remarks in the Office Action regarding the rejection of Claim 24, which also depends from Claim 21, state that Padawer et al. does not explicitly disclose the extension interaction with sub-extension and an extension interface to expose available insert locations to sub-extensions that have commands to insert into an extension UI and to receive a count of command items. Applicants respectfully agree. The remarks go on to state that Ezekiel et al. discloses an iterative process of inter-extension commands and that, therefore, it would have been obvious to one having ordinary skill in the art at the time of this invention to extend an iterative process to inter-extension communications. Claim 24 reads:

> 24. The computer-accessible medium of Claim 21, further comprising:
>
> a declarative list having a command item data structure in which to store an extension command UI, and an insert location data structure to store an available insert location in a extension UI;
>
> an extension interface to expose available insert locations to sub-extensions that have commands to insert into an extension UI, and **to receive a count of command items** and command items representing the sub-extension's commands;
>
> a sub-extension interface to receive the available insert locations from the extension and **to provide the count of command items and command items representing the sub-extension's commands**; and
>
> an extension process to integrate the sub-extension's command into the extension UI prior to **providing to the host the count of command items and command items representing the extension's commands**. (Emphasis added.)

Claim 24 clearly recites providing to the host counts of command items whereas Ezekiel et al. discloses an iterative process. Providing counts is not the same as providing an iterative process. Hence, Ezekiel et al. does not disclose, teach, or remotely suggest the subject matter recited in Claim 24. Further, Ezekiel et al. does not make up for the deficiencies of Padawer et al. with respect to Claim 21 discussed above. Thus, even if it would have been obvious to one having ordinary skill in the art, having the teachings of Padawer et al. and Ezekiel et al. at the time of the invention, to combine the teachings, which applicants deny, the resulting combination would

not meet all of the recitations of Claim 24 combined with Claim 21. As a result, applicants respectfully submit that dependent Claim 24 is allowable.

The Rejection of Claims 9 and 10 Under 35 U.S.C. § 103(a) Based on Padawer et al. and Ezekiel et al. in View of Scouten

The remarks regarding the rejection of Claim 9, which depends from Claim 1, state that Padawer et al. does not disclose a method to keep track of versions as recited in Claim 9. Applicants respectfully agree. The remarks further state that Scouten discloses a system that maps between different versions as recited in lines 24-27 of Col. 1 of Scouten. Applicants respectfully disagree that lines 24-27 of Col. 1 disclose such a system. Lines 24-34 of Col. 1 of Scouten read as follows:

> ...and the macro is applied in turn to each data file.
>
> SUMMARY
>
> According to one aspect of the invention, a method of producing a macro for use by an application program includes producing a macro file including at least two versions of executable configuration code corresponding to two different operating systems to interface the macro file to an application program and attaching to the macro file a sequence of actions to apply to an application file produced by the application program.

Applicants respectfully submit that a macro file that includes at least two versions of executable configuration code is not the same as "remapping the insert locations into a different portion of the host UI than in prior versions of the host UI" as recited in Claim 9. Scouten does not disclose, teach or suggest remapping insert locations for prior versions of a host UI. Further, Scouten does not make up for the deficiencies of Padawar et al. discussed above with respect to Claim 1, the claim from which Claim 9 depends. Hence, even if it would have been obvious to one having ordinary skill in the art at the time of this invention to combine the teachings of Scouten with Padawer et al., which applicants deny, and if Scouten teaches what the Office

Action alleges that Scouten teaches, which applicants also deny, the resulting combination would still not meet the recitations of Claim 9 combined with Claim 1. As a result, applicants respectfully submit that Claim 9 is allowable. Applicants further submit that because Claim 10 depends from Claim 9, Claim 10 is allowable for at least the reasons that Claim 9 is allowable.

CONCLUSION

In view of the foregoing remarks, applicants respectfully submit that all the claims in this application are allowable. Consequently, early and favorable action passing this application to issue is respectfully solicited. If the Examiner has any further questions, the Examiner is invited to contact applicants' attorney at the number set forth below.

Respectfully submitted,

CHRISTENSEN O'CONNOR
JOHNSON KINDNESS PLLC

Gary S. Kindness
Registration No. 22,178
Direct Dial No. 206.695.1702

GSK/MFM:jam

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS PLLC
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100